Efficient Sub-Stream Encoding and Transmission for P2P Video on Demand

Zhengye Liu
Yanming Shen
Shivendra Panwar
Keith W. Ross
Yao Wang

Polytechnic University, Brooklyn, NY, USA

Outline

- Motivation
- P2P VoD system
- Sub-stream encoding and transmission
- Demo
- Conclusion

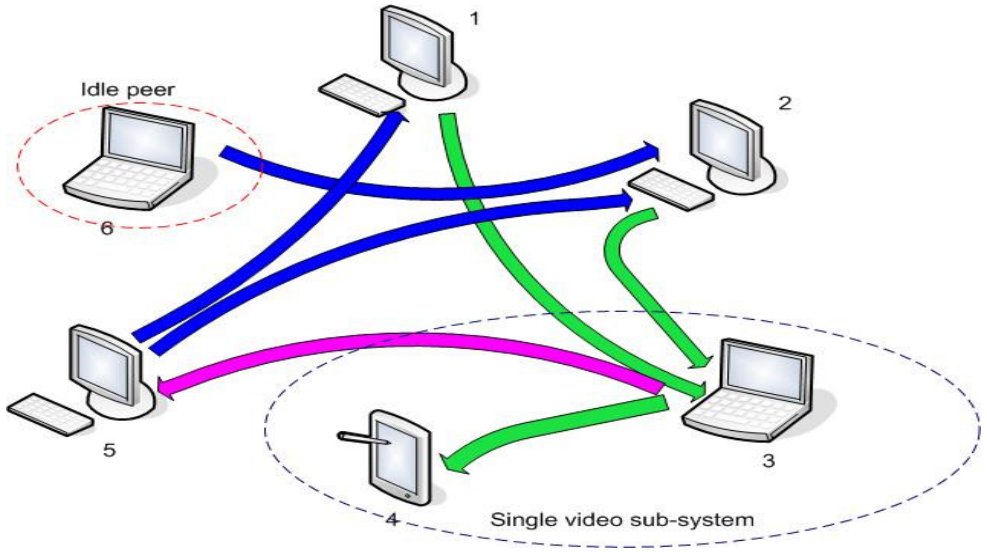
Motivation

- Video-on-demand services
  - Youtube, MSN video, google video,…
  - Content distribution networks (CDNs)

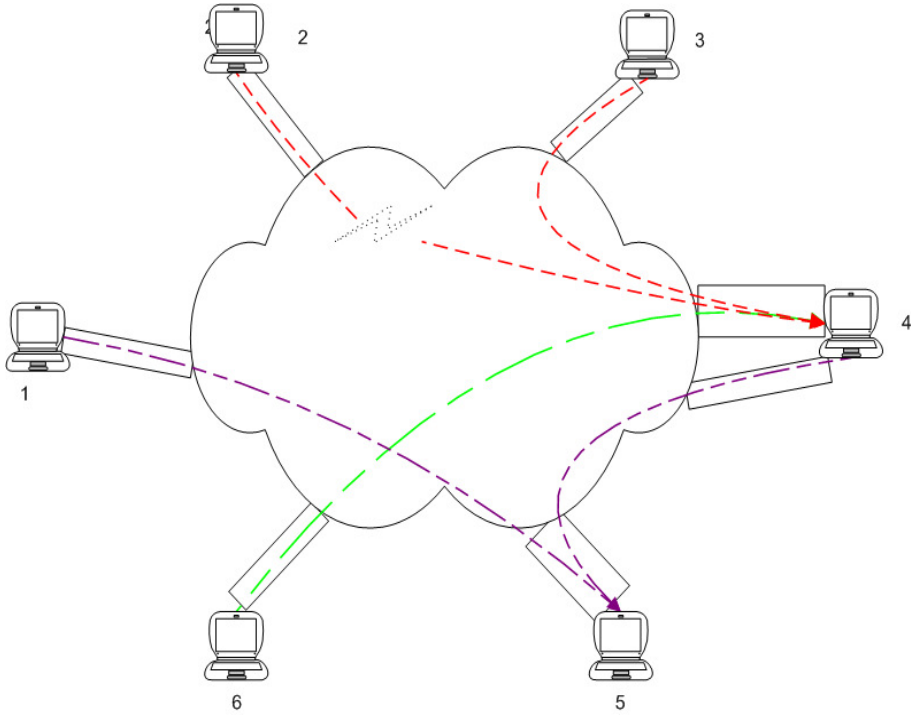
- P2P live streaming systems
  - PPLive, PPStream, UUSee, Coolstreaming,…
  - Support thousands of users simultaneously

P2P VoD System

- Multiple video architecture
  - Extension of CDNs: Peers act as video servers
  - Contribute storage in addition to bandwidth
  - Help with each other with stored videos



Proposed System with Multiple Sub-Streams



In this illustration, two simultaneous streaming sessions are requested from node 4 and node 5. The system initially selects nodes 2 and 3 to serve node 4's request, and select nodes 4 and 1 to serve node 5's request. After Node 2 goes down, the system finds node 6 as a replacement

(a) Layered coding

(b) MDC

(c) Ideal scheme

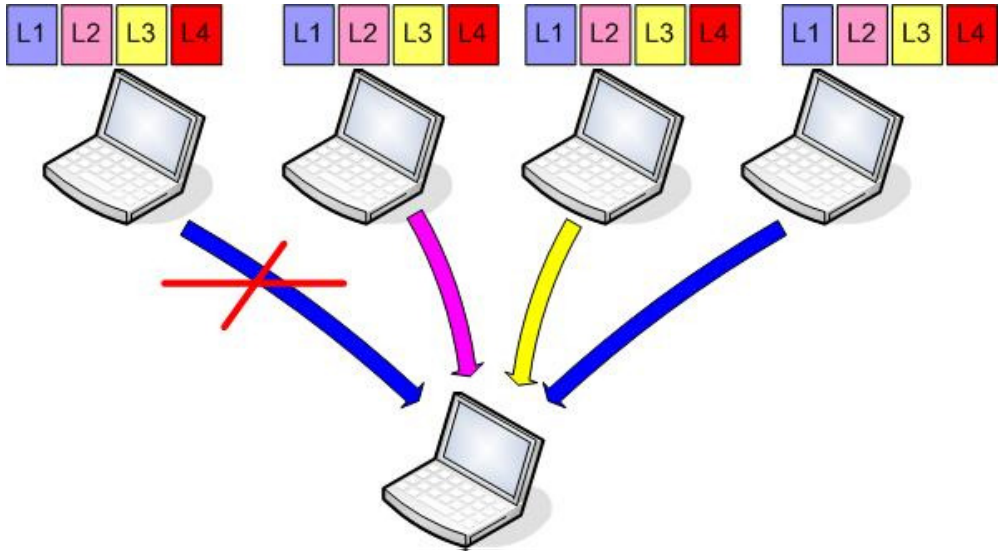
- Compare schemes (a), (b)
- Design (c)

Benefit of Using Layered Coding/MDC

- Adaptive to the long-term bandwidth fluctuation due to peer churn
  - Uplink bandwidth fluctuation
  - Received video quality adapts to the available uplink bandwidth
- Robust to peer failure/disconnection
  - One supplier failure only affects one/several sub-stream(s)
  - Video quality will not be impaired seriously

Push-Pull Delivery with Layered Coding

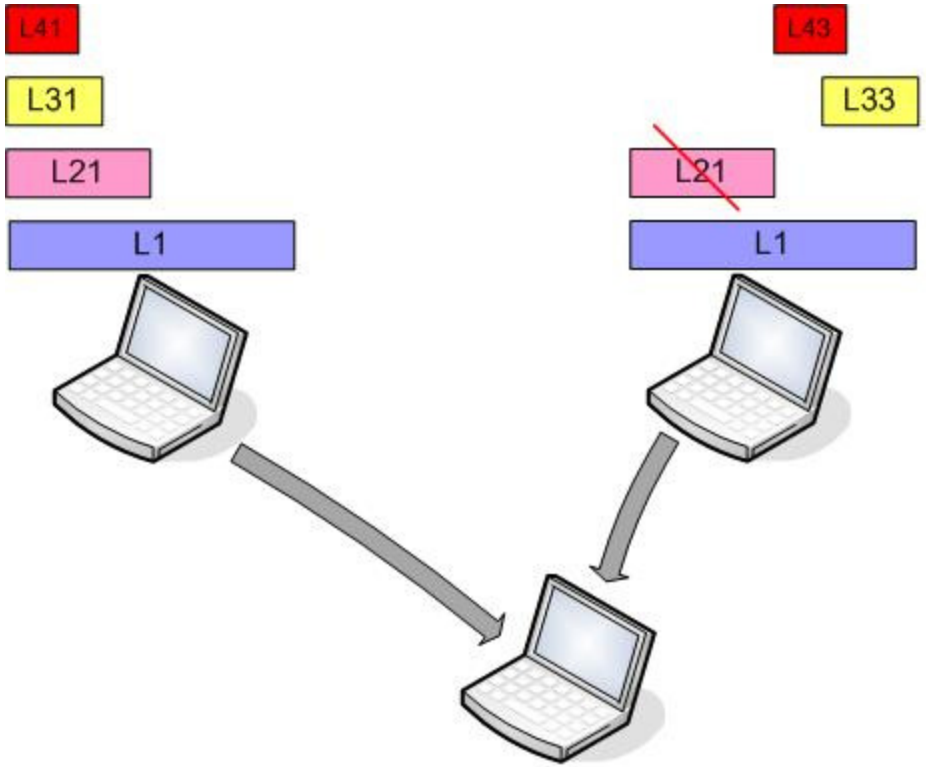
- Store all layers of a video
- Push-pull fashion for layer delivery



- Storage consumed for one video: RM

Minimum Storage Used to Achieve the Ideal Scheme

- Can we reduce the consumed storage?

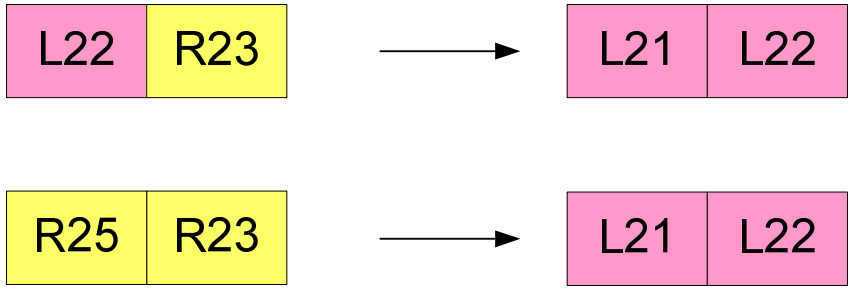


- The minimum storage: $R+R/2+,\ldots,+R/M \approx R \ln(M)$

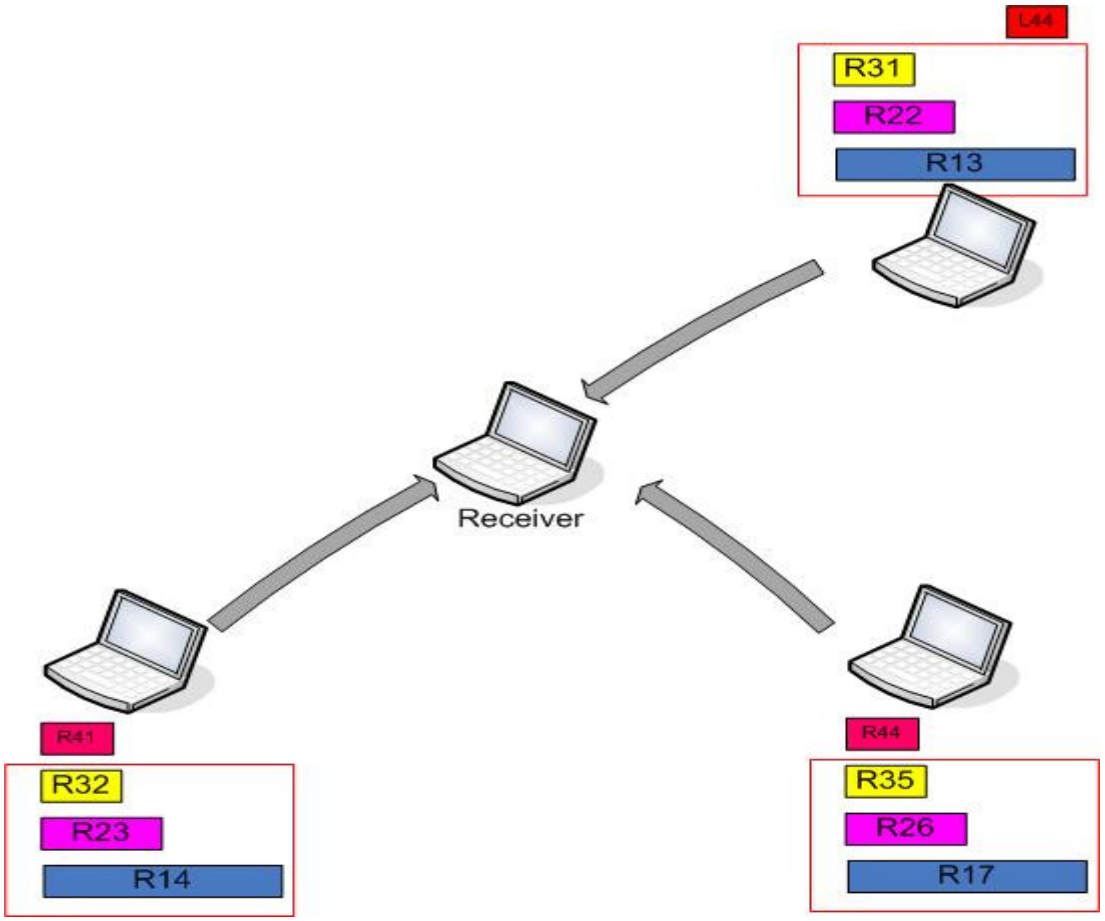
RS Coding

RS (8,2) coding for Layer 2

Any two received chunks can recover the original two chunks

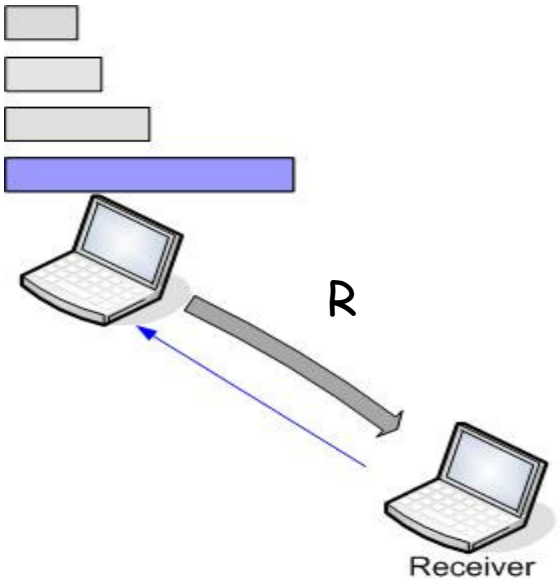


RS Coding Instead of Replicating



RS (8,k) coding,
k=1,2,3,4

Redundancy-Free Transmission Based on Push-Pull Architecture



- A receiver schedules the chunks that should be delivered (Pull)
- A supplier pushes the chunks based on the schedule (Push)

Features of Proposed Scheme

- Ideal Scheme
  - Equal importance (like MDC)
  - Redundancy free transmission (like Layered coding)
- Minimum storage consumed
  - R ln(M) vs. RM
    - M=4, 2.08R vs. 4R, save about 50%
    - M=32, 3.47R vs. 32R, save about 89%

Demo

- Encode a video into 4 layers
- The rate of each layer is 80 kbps
- Use the most recent SVC codec, JSVM

Conclusion and On-Going Work

- Conclusion
  - Propose a P2P VoD system with multiple sub-streams
  - Propose a redundancy-free transmission scheme based on a push-pull architecture
- On-going work
  - Apply scalable video coding on P2P video live streaming, e.g., to provide incentives

Thanks!